Learning Policies of Sim-to-Real and LiDAR-to-mmWave Input Modalities for Collision Avoidance in Subterranean Environments

Jui-Te Huang, Chen-Lung Lu, Po-Kai Chang, Ching-I Huang, Kuan-Lin Chen
Sheng-Cheng Lee, Chao-Chun Hsu, Po-Lin Li, and Hsueh-Cheng Wang*

***Abstract*— Deep reinforcement learning (RL) have shown remarkable success on a variety of tasks to learn from mistakes. To learn collision-free policies for unmanned vehicles, deep RL has been trained with various data modalities including RGB, depth images, LiDAR point clouds without the use of classic map-localize-plan approaches. However, to operate in constrained passages under subterranean environments, existing methods are suffered from degraded sensing conditions, such as smoke and other obscurants, that impairs observations from camera and LiDAR. We propose sim-to-real, LiDAR-to-mmWave (millimeter wave radar) input modality for deep RL to overcome the challenges. We show that the trained models are generalized from simulation to real world, as well as LiDAR to mmWave transferring, despite the low spatial resolution and noisy inputs. Evaluations are carried out in underground environments, including a basement floor and large-scale testbeds in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge. We provide an open dataset of real-world data for further comparisons.**

I. Introduction

Collision avoidance is a fundamental capability for any mobile robots. To achieve it, there have been significant developments in two major topics on simultaneous localization and mapping (SLAM) as well as path planning. Nevertheless, SLAM systems tend to have difficulties with dynamic environments or textureless scenes, and are vulnerable to perception outliers [1]. Path planning requires prior knowledge of the map representation for navigation. Maintaining an accurate map becomes increasingly difficult under poor SLAM results. Moreover, when those classic approaches make a mistake, the encountered failure can not be used to improve the system to prevent the same mistake from occurring in the future.

Navigating in subterranean environments usually encounters degraded sensing, and commonly used sensors such as camera and LiDAR may fail due to the possible adverse smoky or foggy conditions. Millimeter wave (mmWave) radar is a suitable alternative to overcome such challenges of degraded sensing. However, although mmWave radar-based SLAM has been proposed [2], [3], incorporating mmWave inputs for map-localize-plan navigation is still active research problem due to the issues of speckle noise, receiver saturation, multipath reflections, low spatial resolution, and slow update speed. Therefore, map-less navigation via end-to-end solutions from robot sensors to actions are required.

Deep reinforcement learning (RL) algorithms, on the other hand, tackle the problems by learning from trial-and-error experiences. It has been known that training deep RL algorithms are data-intensive, and therefore there have been significant efforts of building simulation for RL training [4]–[8]. Another approach to overcome the data-hunger issue is self-supervised learning [9]–[11] from simulation or real world. Given there exist world-dynamics and perception discrepancies between the simulator and the real world [8], sim-to-real approach only shows limited success under a certain conditions.

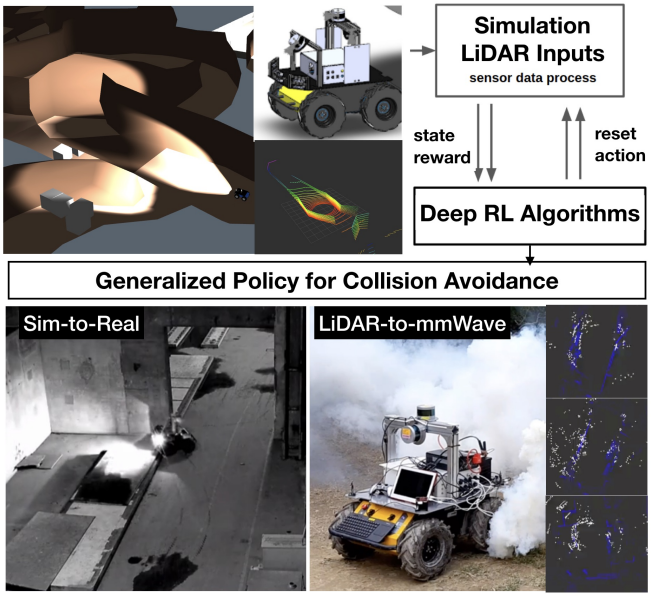

Fig. 1: This work aims at learning a generalized policy of collision avoidance from simulation to real UGV, and from LiDAR to millimeter wave radar modality. The proposed mapless approach, compared to classic map-localize-plan, bypass the needs of mapping, which may be unfeasible in a certain scenarios, and can learn from mistake. We carried out quantitative evaluations in a basement floor and large-scale subterranean environments.

Common data modalities used for deep RL are RGB camera, depth images [12], [13], LiDAR point clouds [14]–[16]. It has been found that range sensing such as depth images or LiDAR inputs are generalized better for sim-to-real transferring than RGB inputs across datasets [12].

We wish to train generalizable policies, and tackle the debating questions of distributional shift between training and test data for deep learning. In specific, are the models trained ? However, it remains unknown about the transferring performances 1) from simulated LiDAR inputs transferred

All authors are with National Chiao Tung University, Taiwan. Corresponding author email: hchengwang@g2.nctu.edu.tw

on embodied real-world robots, and 2) between different data modalities, i.e., from LiDAR to mmWave inputs. To our knowledge this is the first attempt to use mmWave inputs for collision avoidance via map-less deep RL algorithms.

Our work is motivated by the DARPA Subterranean Challenge. In the Tunnel Circuit there have been challenging environments in unknown long coal mine tunnels with smoke emitted adverse conditions that impaired commonly used perception like camera and LiDAR. Classic map-localize-plan may get stuck in a certain corners, which require designs of handcraft state machine as prior. This paper aims at minimizing the handcraft efforts by learning approach, and wish the trained policies generalized for unknown environments. Specifically, this paper aims at tackling the generalization challenges and contributes:

1) **Sim-to-real**: we leverage the use of the SubT Virtual Competition environments [17] and carry out a sim-to-real obstacle avoidance via deep reinforcement learning on UGV. We tested in real-world embodied robots and found robust transferring results for LiDAR inputs.
2) **LiDAR-to-mmWave input modality**: We take advantage of lightweight and low-cost mmWave sensor, which is capable of operating under challenging foggy or smoke-filled environments. We observed performance degradation using mmWave only, due to low spatial resolution.
3) **Evaluations in large-scale subterranean environments**: We carried out experiments for classic map-localize-plan approach, compared to the proposed map-less deep RL approach in subterranean environments, including an indoor hallway, a basement, and in the SubT Tunnel and Urban Circuits. The datasets are openly available [18].

II. Related Work

A. Learning-based Navigation from High-bandwidth Images

There have been significant efforts developing high-quality simulation environments for learning-based navigation. The goal is to provide high-dimensional input (raw images) rather than low-dimension inputs in RL literature. In [4] the training process for policies to navigate was entirely in simulation, and the policies were then deployed and tested in a real-world hallway. The virtual environments were built using the 3D modeling Blender, including rendered images with randomize textures and lighting to create a visually diverse set of scenes. A model of Q-function was used to predict robot actions from camera observation. [5] developed a realistic simulation framework of 3D scenes (AI2-THOR) integrating Unity 3D physics engine. A target-driven visual navigation model was then trained with high-dimensional image inputs to provide end-to-end prediction from pixel information into actions. [6] took advantage of a game engine World Rally Championship 6 (WRC6), and A3C [19] was used to learn car control in a stochastic, realistic environment in self-supervised fashion. The agent took 84x84 front view images and speed as input, and gas, brake, and handbrake as output. Recently Habitat [12] framework was developed with photorealisitics environments Matterport3D [20] and Gibson [21]. A PointGoal navigation [22] task was then carried out by training deep RL model via PPO [23]. Habitat addressed the "embodied AI on enabling actions in environments, rather than in media. Cross-dataset generalization experiments are conducted, and the results suggested depth sensor generalized better across datasets than RGB or RGB-D sensors.

The image to action navigation learning are well-suited for camera-only platform and often used for payload constrained UAVs. [10] trained a CNN model fed by image inputs by collecting an UAV dataset of 11,500 times crashing experiences over 40 hours of real drone flight time with minimal human supervision. Similarly, [11] collected a dataset associating images with collision probability according to the distance to the obstacles, but they trained an UAV to fly from data collected on city streets from the viewpoint of urban [24]. They found the trained policies generalizable, although from bicycles and cars, allowed a drone to fly in indoor corridors and parking lots.

B. Learning-based Navigation from Range Sensing

Different from the image inputs of UAVs, unmanned ground vehicles (UGVs) are able to carry LiDAR for learning-based navigation. Recent work train deep networks either through pre-generated occupancy map to motion commands, or mapless end-to-end approach from range inputs to actions. [7] aimed to train a planner for search and rescue exploration task, given a 64×64 2D local occupancy grid as inputs for neural network. The 2D Stage simulation, usually used for multi-robot problems, was modified in order to train the A3C network for outputs of goal frontier actions. [16] tackled the multi-robot collision avoidance problem in a decentralized scenario. The observation space was obtained from the last three consecutive frames of 512 range values from a 2D laser range finder, resulting in 1,536 dimensions. The actions were in continuous space of translational and rotational velocities. PPO was used to training the model to directly map raw laser input to control outputs from Stage. The results showed that a generalized policy for robots could transverse not only in simulated environment but also real-world environment, even through human-crowded environments. Pfeiffer et al. [14] proposed a data-driven planner to learn motion commands from local geometry obtained by 2D laser range findings. The proposed CNN processed 1,080 dimensions of laser inputs by convolutional layers with two residual building blocks. The fully connected (FC) layers then combined extracted features and the target position. The results showed the model was capable for avoiding obstacles on the road even with unseen objects and reach the final destination. Consequently, a modified neural network was proposed in [15] to downsample 1,080 measurements from LiDAR was into 36 values by a min pooling. Compared to [14], the proposed model was simplified with only FC layers, in order to avoid over-fitting. In [25] only 10-dimensional range findings were used. Such abstract observations reduced the difference obtained from virtual and real environments, and may be feasible for other low-resolution range sensors. Asynchronous deep deterministic

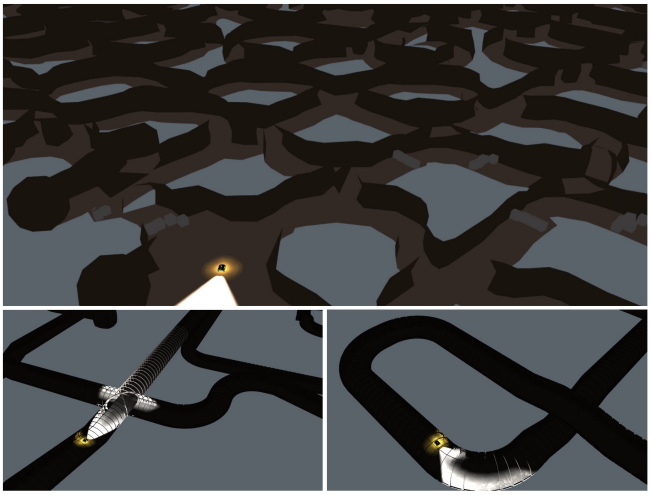

Fig. 2: We use the Virtual SubT environments developed in Gazebo, containing tunnel (bottom), cave (top), and urban environments.

policy gradient (ADDPG) [26], [27] was then used to train the actor and critic models in a V-REP simulator.

We wish to substitute LiDAR inputs into mmWave as range sensing to penetrate environments filled with smoke or fog. We choose the neural network used for range sensing, similar to [14], with convolution and FC layers.

III. Deep Reinforcement Learning for Collision Avoidance

A. Virtual Environment Setup

We train the agent running infinite navigation in subterranean environment using reinforcement learning. Virtual Subterranean [17] Gazebo simulation provide various type of underground environment, include man-made tunnel with obstacle and cliff, natural cave, and subway station. To train a more generalized model, we chose the cave environment which has most diverse size and shape of tunnel and ramp. We follow the OpenAI Gym integrated with the Gazebo simulation to allow process agent's observation data, calculate reward, give actions to, and reset the agent.

We considered DDPG and Recurrent Deterministic Policy Gradient (RDPG) [28] algorithms to train collision avoidance policies from data collected both in Gazebo subterranean simulation environment. Our neural network includes a minimum pooling to downsample, 2 convolution layers to smooth LiDAR inputs, 2 FC layers, 1 recurrent layer with LSTM cell and 1 FC layers to generate action outputs. The network architecture is close to [14] and [16] using 3 consecutive frames, but ours include the LSTM cell. Learning sequential actions by including a recurrent network was found important, which will be discussed in evaluations.

Since the performance of robot learning may heavily depend on the quality and quantity of the collected training data, we investigated the episodes needed to generate training data. We train the model with 495 episodes, each episode contains up to 1,563 steps. We found that at least 450 episodes of off-policy transitions are required to obtain good results in 7 hours training in simulation on a RTX 2070 GPU workstation

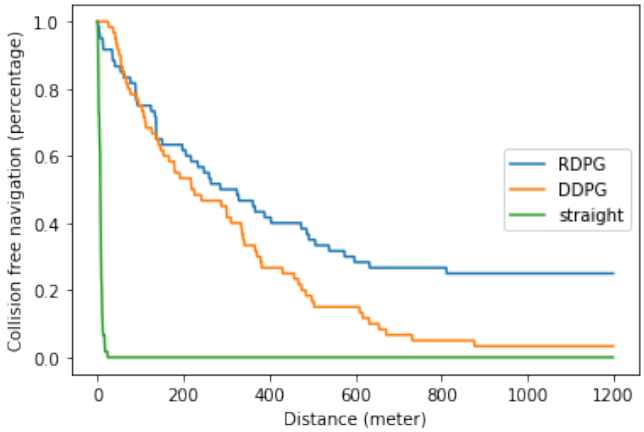

Fig. 3: Quantitative results in simulation. Our approach (DDPG and RDPG) outperformed the straight line baseline. RDPG preformed better than DDPG, indicating the importance of learning from sequential actions.

to speed up the training process, equivalent to approximate 20 hours of training in real world.

The considerations and settings are elaborated as following:

1) Observations: The 3D simulated LiDAR inputs are used for observation space. We sample range findings of angles from -120 degree to 120 degrees, with 1 degree resolution from the point cloud of height a certain height, resulting 241 range values in total. The network was then used for mmWave radar, with lower resolution of 15 degrees, and smoothed by the convolution layers. For mmWave inputs we have a pre-processing stage of filtering to remove outliers by clustering point clouds.

2) Actions: Continuous space of linear and angular velocities are used for actions. The linear velocity ranges from 0 to 1, and the angular velocity is from -1 to 1. The considerations of angular velocity being continuous space over discrete space is to have desired smooth motions, and the linear velocity setting prevents the agent from moving backward.

3) Reward: To encourage agent running infinite navigation. We designed the reward into two parts. First part is aim to encourage agent to go straight and perform less turning action. And the second part is punishment when agent is too close to the obstacle. The reward was normalized to the range from 0 to 1.

$$reward = \begin{cases} 2^{(1-|\omega|)\times 5} & \text{if } min(state) > 1.2 \\ -50 - 300 \times (1.2 - min(state)) & \text{else} \end{cases}$$

where

$$\omega = \text{angular speed of agent}$$

IV. Evaluations

We started with evaluating the performance in simulation, including a baseline and the proposed methods with DDPG and RDPG. From the simulation results we chose RDPG trained solely from simulated source domain to be deployed to embodied real UGV in a 70×25 meters basement. We evaluate variants of our method in different scenarios to understand (a) how the learned policies from simulation to the real world

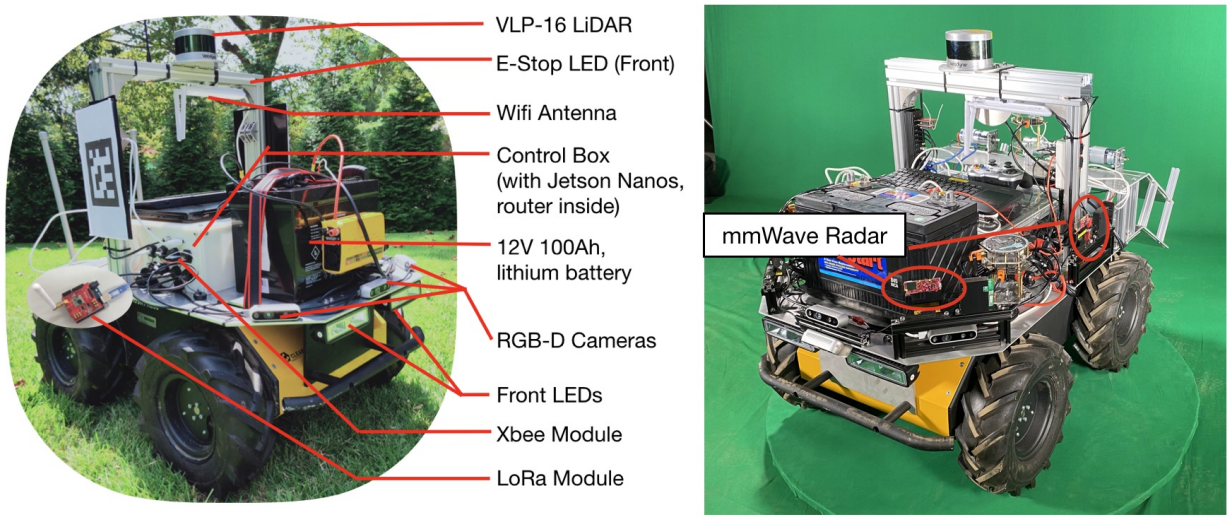

Fig. 4: The UGV Platforms are equipped with Velodyne 3D LiDAR. Left: UGV used in the Tunnel Circuit (LiDAR only). Right: UGV used in the Urban Circuit (LiDAR and mmWave radars).

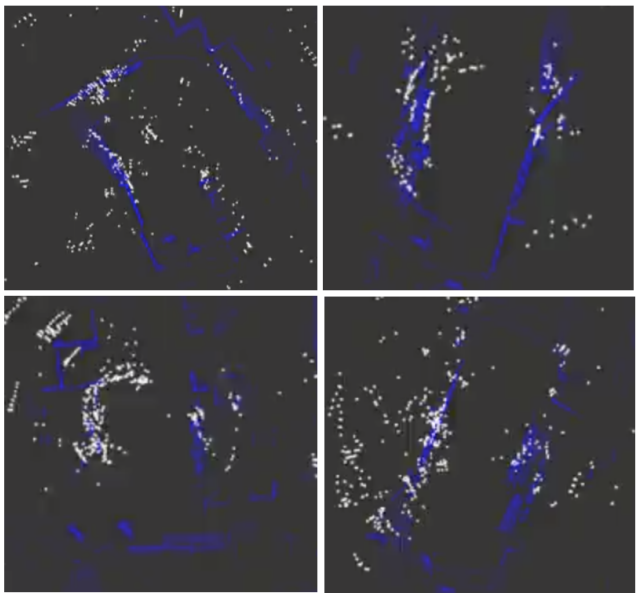

Fig. 5: Top view of the point clouds around the UGV platforms obtained from LiDAR (blue) and mmWave radars (white).

perform, (b) how the real robot performs under different input modalities, and (3) how the full system performs in large-scale subterranean environments.

We evaluated our system in two large-scale subterranean environments. Due to the limited access of the competition site, the numbers of trials were only 2 in the Tunnel Circuit and 1 in the Urban Circuit. Nevertheless, we considered those attempts valuable as comparable results to other participated teams.

We considered a safe navigation without collision a trial. A trial was terminated when the robot collided walls or obstacles, reached to a maximum distance of 1.2 kilometers, or manually stopped by human intervention. The travelled distance was estimated by the wheel encoders provided by the Clearpath Husky robots. Since the environments include rough and muddy floors, in which the measurements may be somewhat biased. The overall results are summarized in Table I.

A. Navigation in Virtual SubT Environments

In the SubT virtual tunnel/cave environments, we initialized 60 trials of random locations with a certain heading. The velocity of the simulated UGV was based on the outputs of the deep network from 0 to 1, and was then scaled to up to 1.5 meter per second. Straight policy is a commonly used weak baseline in previous work. By following a straight path toward a (random) goal point, this baseline will fail in curved paths, collide obstacles on the path and stop. We evaluated DDPG and RDPG, both taking simulated LiDAR as inputs.

As shown in Fig 3, the percentage of collision free navigation of RDPG was higher than the ones of DDPG and straight baseline. About 30 percentages of RDPG trails reached 1.2 kilometers, where as DDPG was suffered from being trapped, especially in dead ends, showing the importance of learning sequential actions. Straight baseline only survived a few meters, since the initial headings were random and there were many turns in the environments.

B. System Platforms

The trained collision avoidance policy from simulation was then deployed on two real UGVs, shown in Fig. 4. Our UGVs were built on Clearpath Husky robots, where we installed required range sensors. The Velodyne VLP-16 LiDAR was set to frame rate to 10 hz, and there were 4 of the 60 GHz TI 6843 mmWave radar modules installed to reach 240 degrees. Fig. 5 showed the point clouds around the UGV platform. Given the noisy point clouds, it is infeasible to carry out classic map-localize-plan approach by mmWave alnoe.

There are many other sensors on the platforms with extra system efforts of managing power supplies and computations. Again the speeds of both robots were computed by the deep network from 0 to 1, but was scaled to up to 0.3 m/s. All ROS bag logs were recorded onboard.

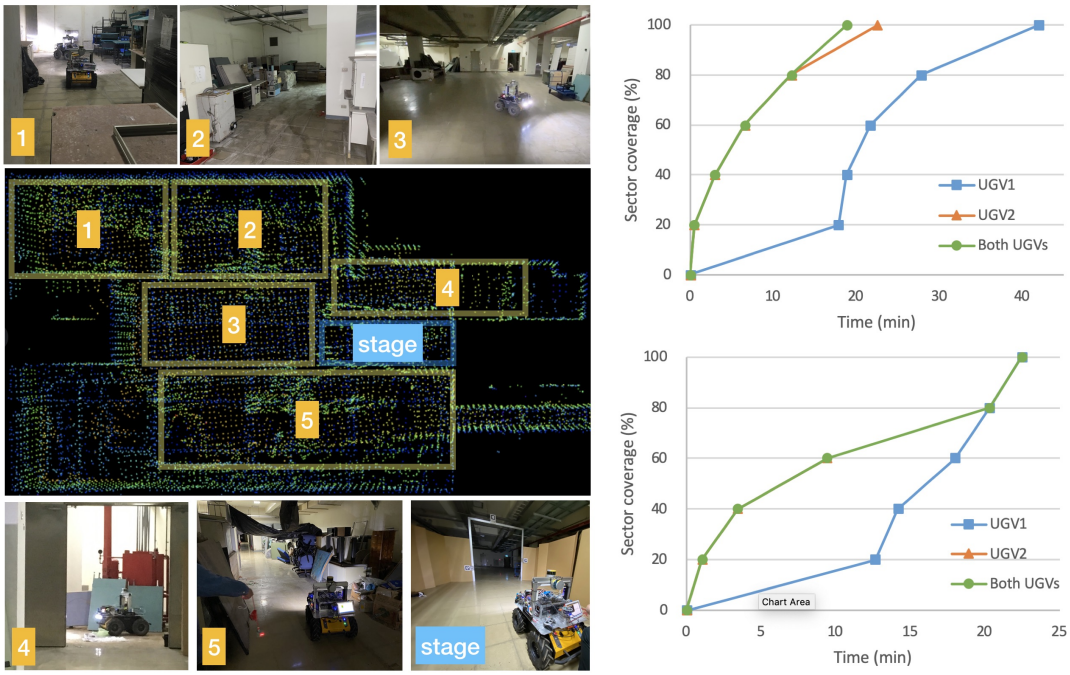

Fig. 6: Left: A real-world cluttered environment in a basement floor. The UGVs started from the stage area (blue), and were expected to navigate through the 5 sectors. Right: two runs of the accumulated sectors reached, each run with two UGVs. The results show that two UGVs reach all 5 sectors sooner than single UGV1 or UGV2.

C. Sim-to-real Results

TABLE I: Experiment Results: average distance in meters (Dist) and average time in seconds (Time) before collision and before stop that requires human intervention.

	Total Col./Stop	Avg. Time	Avg. Dist.	Max Single Safe Dist.
EF-B1 Testing Env.				
RDPG - LiDAR	7	32.25	407.62	675.41
RDPG - mmWave	14	2.20	30.83	-
Large-Scale SubT Env.				
Urban Circuit - RDPG	1	17.12	277.53	277.53
Tunnel Circuit - Baseline	2	15.77	193.70	236.74

The testing real environment was a basement floor in the EF Building at National Chiao Tung University. There were open space and several pathways with several turns in narrow corridors, cluttered with partitions, tables, chairs, furniture, and other obstacles. There were also some artifacts, such as red backpack or survivor manikin, placed in the environments. The UGVs were initialized at the staging area, and they were expected to maneuver through the 5 sectors connected by the central open space sector 3, shown in Fig. 6. We carried out 2 runs of two UGVs operating at the same time. We evaluated the accumulated coverage of sector reached by the two robots. In general two UGVs could reach all sectors sooner. We sent first UGV to enter sectors in the beginning of the run, and subsequently the second UGV about 10 minutes later. Therefore the coverage of both UGVs was closer to the one of UGV entered first. In both runs sector 1-5 were reached around 20 minutes of exploration.

Table I showed the UGVs could reach more than 400 meters in average, indicating the policy trained in simulation were successfully deployed and transferred on physical robots. We observed only 1 collision in sector 1 (shown in Fig. 6), caused by two UGVs operating in a crowded narrow dead end. Notice that our policy was only trained in a single robot scenario, and both robots were able to avoid collisions to each other and surrounding obstacles, except in a narrow dead end.

D. LiDAR-to-mmWave Results

As shown in Table I, the performance of mmWave inputs suffers a significant loss compared to one of LiDAR inputs. Nevertheless, the UGV could reach 30 meters safe navigation in average carrying out mapless navigation from learning approaches. Given that the point clouds from mmWave only appear while the UGV was moving, the model sometimes carried out sweeping in a certain angles with 0 linear velocity. Among the collisions the UGV usually hit knee-height obstacles, this may be due to the limitations of low spatial resolution. The slow updating rate may also affect the performance. We still consider such performance useful, especially traverse through smoke-emitted environments.

E. Large-scale Subterranean Environments

1) Baseline Approach in the Tunnel Circuit: . The Tunnel Circuit was held at NIOSH, Pittsburgh, including two mine tunnels (Experimental Mine and Safety Research Mine) were maintained for research purposes. The tunnels extended 2-4 kilometers in length and included constrained passages. We implemented a baseline map-localize-plan tunnel following

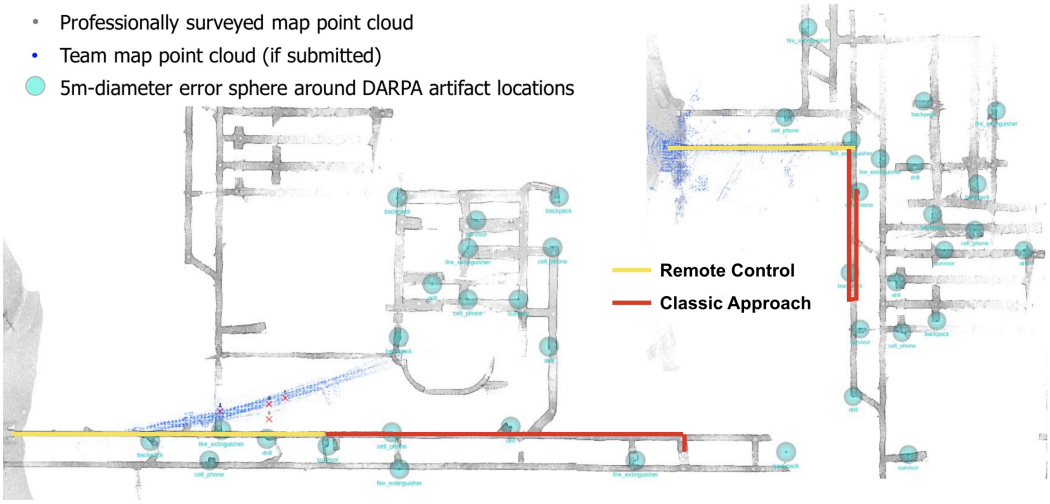

Fig. 7: Distance travelled results using classic map-localize-plan baseline approach in the Tunnel Circuit of the SubT Challenge. Left: Experimental Mine. Right: Safety Research Mine.

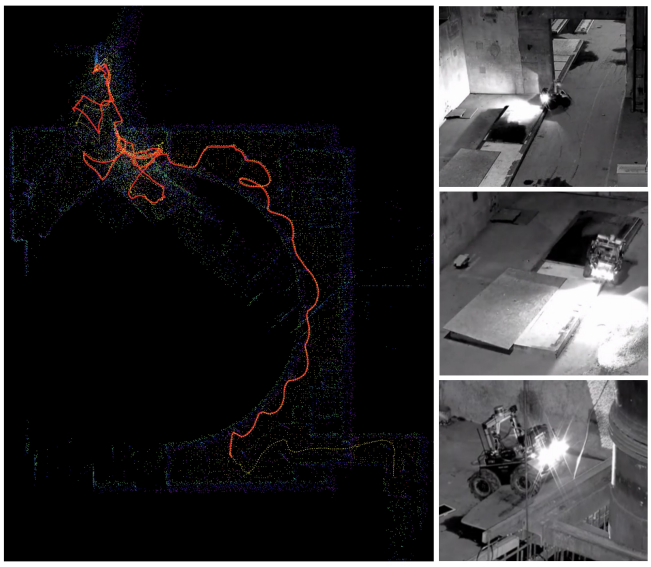

Fig. 8: Distance travelled results using the proposed approach in the Urban Circuit of the SubT Challenge. The trajectories showed that the proposed approach carried out more turns to avoid obstacles and travelled longer in the Urban Circuit than the baseline in the Tunnel Circuit.

policy to perform exploration. The robot state at time t was represented as $x_t = \langle d_t, \phi_t \rangle$, where d_t was the lateral distance between the robot and the center of the tunnel at time t and ϕ_t is the angle relative to the tunnel axis. By parsing the point cloud gathered from the RGB-D or LiDAR inputs, we built an occupancy grid map. The map was then used for a local planner, sweeping from -90 to 90 degrees with a 2-meter radius. A goal point was then set to the found traversible point via A* search. Such settings tended to keep the robot staying at a certain lateral distance with $\phi = 0$. Nevertheless, this baseline method may not handle well open space, intersections, or dead end situations. Table I demonstrated the two attempts of autonomous runs were carried out in the two courses, shown in Fig. 7. In the Experimental Mine the robot travelled 237 meters, and the robot was trapped in a dead end. In the Safety Research Mine, due to the communication limitation we set the robot to traverse through a certain distance and return to communication enabled area for artifact report, and the travelled distance was 149 meters.

2) The Proposed Approach in the Urban Circuit: .

The Urban Circuit took place at the Satsop Business Park in Elma, Washington. The environments include localization-unfriendly flat surfaces, dirt, water, stairs, and ledges. There were two courses, the Alpha and Beta Courses, and Fig. 8 demonstrated the robot trajectory in the Alpha Course. We obtained 277.53 meters from wheel odometry by the use of the proposed approach. The robot were trapped on a ledge due to the limitation of downsampled LiDAR inputs on different height levels. In the Beta Course, the trained policy was too conservative to pass through narrow passages in the beginning of the course, and the robot circled around in an open area without exploration.

V. Conclusions

Inspired by the DARPA SubT Challenge, we developed a mapless collision avoiding method, which successfully transferring policies from simulated to real robots, and from LiDAR to mmWave inputs for subterranean search and rescue missions. We carried out experiments in real-world basement and large-scale environments in the Tunnel and Urban Circuits. To our knowledge, this work is the first attempt for evaluating LiDAR-to-mmWave input modality that carry out mapless navigation from mmWave inputs. Different from existing work evaluated in small-scale laboratory environments, our quantitative results in large-scale subterranean environments demonstrated hundred of meters safe navigation in complex environments.

However, we observed conservative policies to narrow corridors, which may be further finetuned. The current policy is also vulnerable to ledge or negative obstacles due to the use of downsampled LiDAR inputs. Such challenges may be overcomed by high-bandwidth depth image inputs

and corresponding deep network structures. Although the performance of the mmWave inputs showed some success, further work is needed to train the policy with mmWave inputs from real-world collision experience. We have also investigated the generalizability of the policy trained from UGV to UAV. Nevertheless, we did not obtain good results due to the different dynamics of robots. We will continue investigating the policies transferring among robot modality in the future.

ACKNOWLEDGMENTS

We are thankful for the help from Po-Jui Huang, Tsai-Yo Ting, and Yi-Hsuan Huang. The research was supported by Ministry of Science and Technology, Taiwan (grant 107-2923-E009-004-MY3, 108-2218-E-007-039, and 108-2321-B-009-004). This work was funded in part by Qualcomm through a Taiwan University Research Collaboration Project.

REFERENCES

[1] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-slam: Distributed, online, and outlier resilient slam for robotic teams," *arXiv preprint arXiv:1909.12198*, 2019.

[2] S. H. Cen and P. Newman, "Radar-only ego-motion estimation in difficult settings via graph matching," *arXiv preprint arXiv:1904.11476*, 2019.

[3] ——, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[4] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[5] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[6] M. Jaritz, R. de Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2070–2075.

[7] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[8] H. Bharadhwaj, Z. Wang, Y. Bengio, and L. Paull, "A data-efficient framework for training and sim-to-real transfer of navigation policies," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 782–788.

[9] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[10] D. Gandhi, L. Pinto, and A. Gupta, "Learning to fly by crashing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 3948–3955.

[11] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[12] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[13] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2371–2378.

[14] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 ieee international conference on robotics and automation (icra)*. IEEE, 2017, pp. 1527–1533.

[15] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.

[16] T. Fan, P. Long, W. Liu, and J. Pan, "Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios," *arXiv preprint arXiv:1808.03841*, 2018.

[17] O. Robotics. (2016) Darpa subterranean challenge virtual competition. [Online]. Available: https://bitbucket.org/osrf/subt/wiki/Home

[18] H.-C. Wang. (2020) Datasets for learning policies of sim-to-real and lidar-to-mmwave inputmodalities for collision avoidance in subterranean environments. [Online]. Available: https://arg-nctu.github.io/projects/deeprl-ugv.html

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[20] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[21] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.

[22] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] Udacity. (2016) An open source self-driving car. [Online]. Available: https://www.udacity.com/self-driving-car

[25] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning." in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15

[27] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[28] N. M. O. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *ArXiv*, vol. abs/1512.04455, 2015.